

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) An integrated tracing and logging system employed within a network comprising:

a computer system having:

a processor coupled with a memory, ~~the computer system further including,~~

a tracing module associated with specified program code regions of an application, the tracing module to receive via an application programming interface (API) and process tracing method calls generated by the application when the specified program code regions are executed,

a logging module separate from the tracing module, the logging module associated with specified categories related to the network, the logging module to receive via the API and process logging method calls from network components associated with the categories, and

a formatter coupled to the tracing module and to the logging module, the formatter to receive ~~an output~~ tracing messages from ~~one of the~~ tracing module and logging messages from the logging module, the formatter including a configuration file storing a default format definition for the formatter, the formatter initially to format messages from the tracing module and the logging module according to the default format definition during a runtime of the formatter, wherein the formatter further to be reconfigured during the runtime to format messages from the tracing module and the logging module according to a changed format definition, the reconfiguring the formatter including the configuration file of the formatter further to receive receiving a change to the default format definition for the formatter during a runtime of the integrated tracing and logging system, wherein receiving the change to the format definition for the formatter during the runtime, wherein the reconfiguring the formatter does not require recompiling of any source code of the integrated

tracing and logging system, ~~wherein the formatter to format the output from the one of the tracing module and the logging module according to the changed format definition;~~ and

an output destination to receive the formatted output of ~~the one of~~ the tracing module and the logging module.

2. (Original) The system of claim 1, wherein the formatter is one of a list formatter, a human-readable formatter, and a markup language formatter.

3. (Currently Amended) The system of claim 1, wherein ~~one or more properties of the formatter are defined in the configuration file~~ further defines one or more properties of the formatter.

4. (Original) The system of claim 3, wherein the configuration file includes an identifier to identify the formatter.

5. (Original) The system of claim 3, wherein the one or more properties are formatted as key-value-pair properties, each key-value pair having a key to specify an attribute and a value to provide a definition for the specified attribute.

6. (Currently Amended) The system of claim 3, wherein a format definition of the configuration file ~~defines the message format for the received message, the message format including~~ includes one or more fields.

7. (Currently Amended) The system of claim 6, wherein the one or more fields ~~of the message format~~ includes at least one of

a timestamp field to indicate a time for the received message;

a location of origin field to indicate a source of the received message;
a thread identifier field to indicate a thread associated with the received message;
a message severity indicator field to indicate a severity level of the received message;
and
a message identifier field to identify the received message.

8. (Currently Amended) The method of claim 1, wherein the output destination is at least one of

a console;
a trace file; and
a log file.

9. (Currently Amended) The method of claim 1, wherein ~~the output destination is a console~~ the logging module and the tracing module are different respective subclasses of a controller class, wherein the output destination is inherited from the controller class both by the logging module and by the tracing module, and wherein the formatter receiving the tracing messages from the tracing module and the logging messages from the logging module is based on an assignment of the formatter to the output destination.

10. (Currently Amended) A computer-implemented method employed within a network comprising:

creating an instance of a tracing controller associated with specified program code regions of an application, the tracing controller instance to receive and process tracing method calls generated by the application when the specified program code regions are executed;

creating an instance of a logging controller associated with specified categories related to the network, the logging controller to receive and process logging method calls from network components associated with the categories;

providing a common application programming interface of the tracing controller instance and the logging controller instance, whereby the tracing controller instance and the logging controller instance are accessed;

creating an instance of a formatter coupled to the tracing ~~module~~ controller instance and the logging ~~module~~ controller instance, the formatter including a configuration file ~~storing a format definition for the formatter; after the creating the instance of the formatter;~~

receiving at the formatter instance tracing messages from the tracing controller instance and logging messages from the logging controller instance;

the formatter instance performing a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a default format definition stored in the configuration file;

after the first formatting, reconfiguring the formatter instance to format messages from the tracing controller instance and the logging controller instance according to a changed format definition, the reconfiguring during the runtime, the reconfiguring including changing the default format definition stored in the configuration file, wherein ~~changing the format definition~~ reconfiguring the formatter instance does not require a recompiling of any source code; ~~receiving at the formatter an output from one of the tracing module and the logging module; and formatting at~~

after the reconfiguring, the formatter ~~the output~~ instance performing a second formatting of received messages from the one of the tracing module and the logging module, the second formatting according to the changed format definition ~~of the configuration file.~~

11. (Original) The method of claim 10, further comprising:
configuring the message format for the selected formatter.

12. (Original) The method of claim 11, wherein configuring the message format comprises providing an identifier to the configuration file to identify the selected formatter.

13. (Original) The method of claim 12, wherein configuring the message format further comprises specifying one or more fields for the message format.

14. (Original) The method of claim 13, wherein specifying one or more fields comprises specifying at least one of

a timestamp field to indicate a time for the received message;

a location of origin field to indicate a source of the received message;

a thread identifier field to indicate a thread associated with the received message;

a message severity indicator field to indicate a severity level of the received message;

and

a message identifier field to identify the received message.

15. (Original) The method of claim 10, further comprising:

providing a filter to the specified output destination to selectively filter the message.

16. (Currently Amended) A system comprising:

a computer system having a processor coupled with a memory, the computer system further including,

a means for creating an instance of a tracing controller associated with specified program code regions of an application, the tracing controller instance to receive via an application programming interface (API) and process tracing method

calls generated by the application when the specified program code regions are executed,

a means for creating an instance of a logging controller associated with specified categories related to the network, the logging controller to receive via the API and process logging method calls from network components associated with the categories,

means for creating an instance of a formatter to receive ~~an output from one of~~ tracing messages from the tracing controller instance and logging messages from the logging controller instance, the formatter instance including a configuration file ~~storing a format definition for the formatter instance,~~ the formatter instance further to perform a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a format definition stored in the configuration file, and

means for reconfiguring the formatter instance after the first formatting and during the runtime of the formatter instance, the reconfiguring including changing the format definition stored in the configuration file ~~for the formatter instance,~~ wherein the ~~changing the format definition~~ reconfiguring the formatter instance does not require a recompiling of any source code, wherein the reconfigured formatter instance ~~further to format the output to perform a second formatting of received messages~~ from ~~the one of~~ the tracing controller instance and the logging controller instance, the second formatting according to the changed format definition stored in the configuration file; and

an output destination to receive the formatted output of the one of the tracing controller instance and the logging controller instance.

Claims 17. (Canceled).

18. (Previously Presented) The system of claim 16, wherein the means for changing the format definition comprises:

a means for specifying one or more fields for a defined message format.

19. (Previously Presented) The system of claim 18, wherein the means for specifying one or more fields comprises a means for specifying at least one of

a timestamp field to indicate a time for the received output;

a location of origin field to indicate a source of the received output;

a thread identifier field to indicate a thread associated with the received output;

a message severity indicator field to indicate a severity level of the received output;

and

a message identifier field to identify the received output.

20. (Currently Amended) An article of manufacture comprising:

an electronically accessible medium providing instructions that, when executed by an apparatus, cause the apparatus to

create an instance of a tracing controller associated with specified program code regions of an application, the tracing controller instance to receive and process tracing method calls generated by the application when the specified program code regions are executed;

create an instance of a logging controller associated with specified categories related to the network, the logging controller to receive and process logging method calls from network components associated with the categories;

provide a common application programming interface of the tracing controller instance and the logging controller instance, whereby the tracing controller instance and the logging controller instance are accessed;

create an instance of a formatter coupled to the tracing ~~module~~ controller instance and the logging ~~module~~ controller instance, the formatter including a configuration file ~~storing a format definition for the formatter~~; after the creating the instance of the formatter, change;

receive at the formatter instance tracing messages from the tracing controller instance and logging messages from the logging controller instance;

perform at the formatter instance a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a default format definition stored in the configuration file;

after the first formatting, reconfigure the formatter instance to format messages from the tracing controller instance and the logging controller instance according to a changed format definition, the reconfiguring during the runtime, the reconfiguring including changing the default format definition stored in the configuration file, wherein changing the format definition reconfiguring the formatter instance does not require a recompiling of any source code; ~~receiving at the formatter an output from one of the tracing module and the logging module; and format at~~

after the reconfiguring, the formatter the output instance performing a second formatting of received messages from the one of the tracing module and the logging module, the second formatting according to the changed format definition of the configuration file.

Claim 21. (Canceled).

22. (Currently Amended) The article of manufacture of claim 20, wherein the instructions that, when executed by the apparatus, cause the apparatus to change the format definition for the formatter, cause the apparatus to provide one or more fields for a defined message format.

23. (Currently Amended) An apparatus comprising:

an application; and

a processor and logic executable thereon to

create an instance of a tracing controller associated with specified program code regions of the application, the tracing controller instance to receive and process tracing method calls generated by the application when the specified program code regions are executed;

create an instance of a logging controller associated with specified categories related to a network, the logging controller instance to receive and process logging method calls from network components associated with the categories, wherein the logging controller instance and the tracing controller instance are different respective subclasses of a controller class, wherein an output destination is inherited from the controller class both by the logging controller instance and by the tracing controller instance;

provide a common application programming interface of the tracing controller instance and the logging controller instance, whereby the tracing controller instance and the logging controller instance are accessed;

create an instance of a formatter coupled to the tracing ~~module~~ controller instance and the logging ~~module~~ controller instance, the formatter including a configuration file ~~storing a format definition for the formatter; after the creating the instance of the formatter, change,~~ wherein the formatter is one of a list formatter, a human-readable formatter, and a markup language formatter;

receive at the formatter instance tracing messages from the tracing controller instance and logging messages from the logging controller instance, the receiving based on an assignment of the formatter instance to the output destination;

perform at the formatter instance a first formatting of received messages from the tracing controller instance and the logging controller instance during a runtime of the formatter instance, the first formatting according to a default format definition stored in the configuration file;

after the first formatting, reconfigure the formatter instance to format messages from the tracing controller instance and the logging controller instance according to a changed format definition, the reconfiguring during the runtime, the

reconfiguring including changing the default format definition stored in the configuration file, the changing the default format definition including specifying one or more fields for a defined message format, the one or more fields including at least one of

a timestamp field to indicate a time for the received message,

a location of origin field to indicate a source of the received message,

a thread identifier field to indicate a thread associated with the received message,

a message severity indicator field to indicate a severity level of the received message, and

a message identifier field to identify the received message,

wherein ~~changing the format definition~~ reconfiguring the formatter instance does not require a recompiling of any source code; ~~receiving at the formatter a message from one of the tracing module and the logging module;~~ and format at

after the reconfiguring, the formatter the message instance performing a second formatting of received messages from the one of the tracing module and the logging module, the second formatting according to the changed format definition of the configuration file.

24. (Canceled).

25. (Original) The apparatus of claim 23, wherein the configuration file includes an identifier to identify the formatter.

Claims 26 through 28. (Canceled).